Getting around in vi

There are two modes in vi: command and insert modes. You insert text during insert mode, and issue commands in command mode. Hitting the <escape> key will always take you out of insert mode and put you into command mode. If you were already in command mode when you hit <esc>, it will merely beep at you and you will remain in command mode.

Remember that vi is case sensitive, and therefore j is a different command from J. If you see unexpected things happening in response to your commands, check to make sure you don't have "Caps Lock" selected on the key board.

vi <filename>	vi's the file for editing or viewing
view <filename>	opens the file for vi'ing in Read-Only mode. This is safer and should be used whenever you are merely looking at, and not planning on changing, a file.

Commands to move around, in command mode:

j	move down one line
k	move up one line
l	move left one character
h	move right one character
:0	move to the first line (go to the top of the file)
:50	move to the 50th line
G	move to the bottom of the file
w	move to the next word beginning
b	move to the previous word beginning
<Control f>	moves forward a page
<Control b>	moves backward a page

Commands to change text, in command mode:

dd	deletes current line (current line is the line the cursor is on)
dw	deletes the current word
cw	changes the current word. It will put you in inset mode and show "$" at the end of the current word. Type the replacement text, and hit <esc> when you are done. This will insert the relacement text between the beginning of the current word and the "$". The size of the replacement text does not need to equal or be less than the size of the word that you replaced.

x	deletes the current character
r<character>	replaces current character with the character you type after the "r". Note this leaves you in command mode.
R<characters>	Replaces from the current character onward. Note this leaves you in insert mode until you hit <esc>.
u	undos last command (very useful)

Commands to cut and paste

(Use Yank and Put to copy; Delete and put to cut)

yy	yanks current line into the anonymous buffer
nyy	yanks n lines from current line down into the anonymous buffer (e.g., 5yy yanks 5 lines)
p	"puts" whatever is in the buffer below the line you are currently on
P	"puts" whatever is in the buffer above the line you are currently on

Note: put works with anything in the buffer. If you deleted a line or word (dd or dw) or character (x) or multiple lines, words or characters, they are temporarily stored in the anonymous buffer. P or p will get them back from the buffer.

mj	puts a marker of "j" at the place where the cursor is. You will not see the "j." You could mark with any letter (e.g., ma, md, mq). This is used to mark the END of a block for copying or cutting and then pasting.
'j	(Single quote and then the marker letter). moves the cursor to the marker j. Could be useful for moving around a large document.
d'j	deletes from the cursor to the marker j. Puts contents into the anonymous buffer.
y'j	copies from the cursor to the marker j. Puts contents into the anonymous buffer.
p	puts contents from the anonymous buffer after the line the cursor is on
P	puts contents from the anonymous buffer before the line the cursor is on

Note: for named buffers, where the name of the buffer is a letter a-z, precede the above commands that involve the buffer with a double quote and the letter of the buffer. For example, for buffer b, deleting from the cursor until the marker j:

“bd’j	deletes from the cursor to the marker j. Puts contents into the b buffer.
“bp	puts contents from the b buffer after the line the cursor is on

Repeating commands: <#> then command results in that command done that number of times:

examples (in command mode):

5j	moves down 5 lines
100k	moves up 100 lines
45dd	deletes 45 lines (the one you’re on and the next 44)
(then u)	undos the delete and brings all 45 lines back
3x	deletes 3 characters (the on you’re on and the next 2)

To insert text, you must be in insert mode. The following puts you into insert mode. Again, <esc> gets you out of input mode and into command mode (above are commands you do in command mode). If unsure if you are in command or input mode, hit escape, and regardless of which you were in, you will be put into command mode (if you HAD been in command mode, it will beep at you).

To get into insert mode:

i	insert at current character (then just type in what you want, exc to get out of inserting)
o	inserts following current line (then type, then <esc>)
O	inserts prior to current line (then type, then <esc>)
a	appends. Inserts after current character (then type, then <esc>)
A	appends at end of line. Inserts after last character of line (then type, then <esc>).

Searching in command mode:

/<pattern>	forward searches for pattern
?<pattern>	backward searches for pattern

In both of the above, the command will move the cursor to the next/prior occurrence of the pattern.

n	finds the next occurrence of <pattern>, whichever search you did last.

Replace ALL instances of <old pattern> with <new pattern> in the file.

:g/<old pattern>/s//<new pattern>/g

To replace ALL instances of <old pattern> with nothing, in effect removing all instances of old pattern, do:

:g/<old pattern>/s///g

To save and exit:

:w	saves the file as current filename (and therefore writes OVER the old version of the filename), and keeps you in vi.
:w <new filename>	same as above, but if filename is different, will save current version under that name, not changing original file.
:q	quits. If you have made not changes, this will let you exit from vi. If you have made changes, it prompts you that changes will not be saved. To exit without saving after making changes, you must type :q!
:q!	quits whitout saving changes (this is good if you mess something up). Original file is unchanged.
:wq	writes (saves) to original filename, and quits.

Useful commands:

:r <filename>	This imports the named file onto the line following the cursor. The file is inserted, so the original file below the corsor is moved to be following the inserted file. This is useful in mail.
. (period)	This will repeat the last issued command. Very useful.
<Control g>	Gives information about the file at the bottom of the screen. It will indicate the filename, what line number the cursor is on out of how many lines total in the file, and the percent of the way through the file that the cursor is.